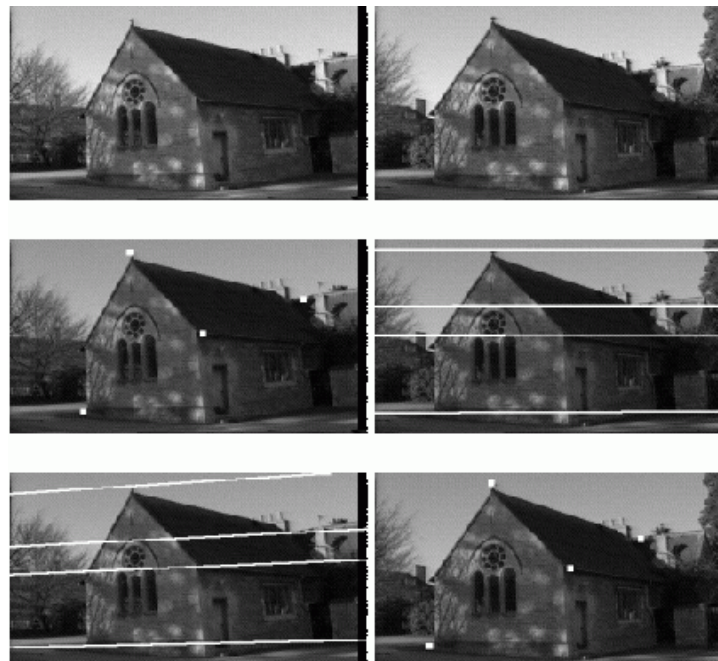


CS 4495 Computer Vision

N-Views (2) – Essential and Fundamental Matrices

Aaron Bobick
School of Interactive Computing



Administrivia

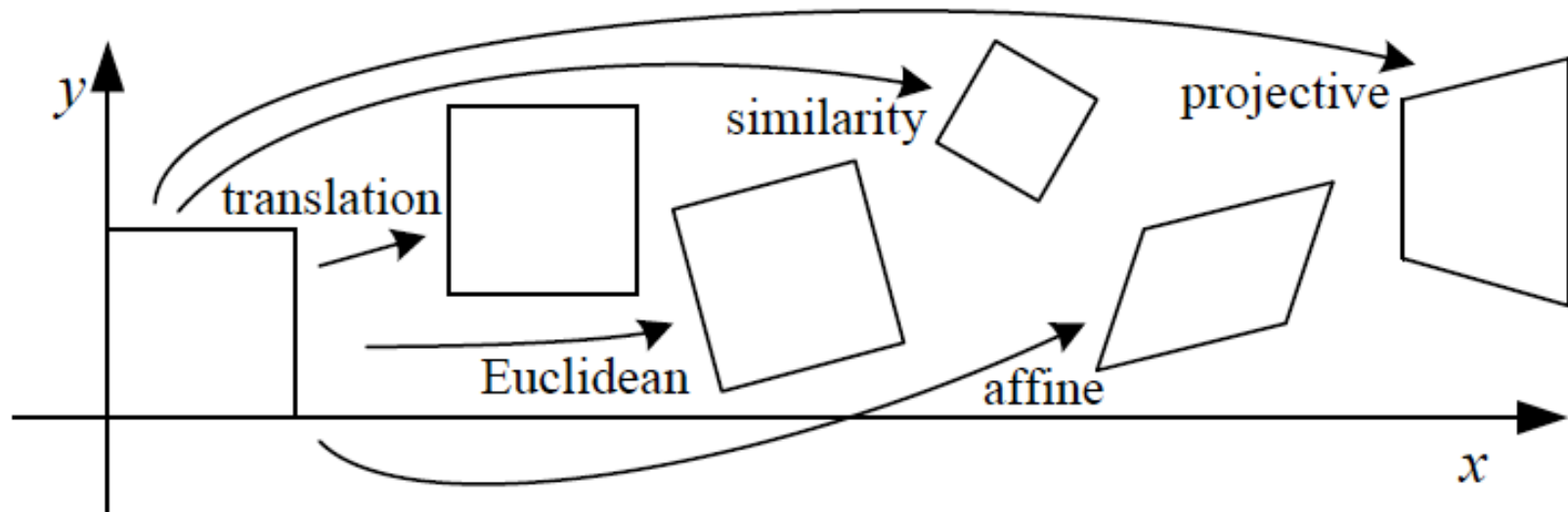
- Today: Second half of N-Views ($n = 2$)
- PS 3: Out tomorrow.
 - Will be due October 12th.
 - Will be based upon last week and today's material

Two views...and two lectures

- Projective transforms from image to image
- Some more projective geometry
 - Points and lines and planes
- Two arbitrary views of the same scene
 - Calibrated – “Essential Matrix”
 - Two uncalibrated cameras “Fundamental Matrix”
 - Gives epipolar lines

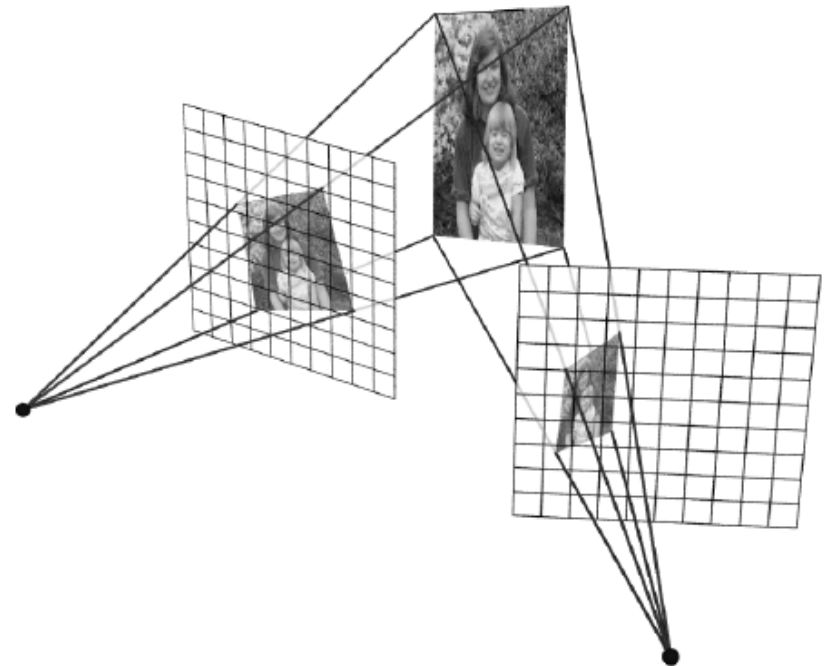
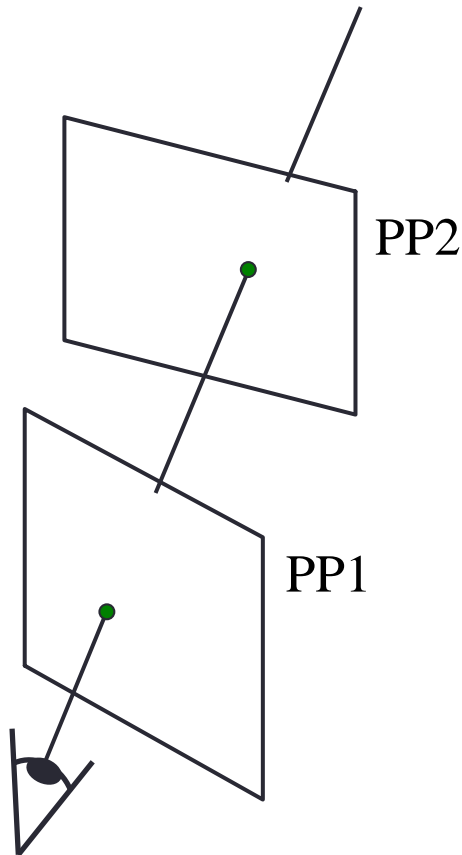
Last time

- Projective Transforms: Matrices that provide transformations including translations, rotations, similarity, affine and finally general (or perspective) projection.
- When 2D matrices are 3×3 ; for 3D they are 4×4 .

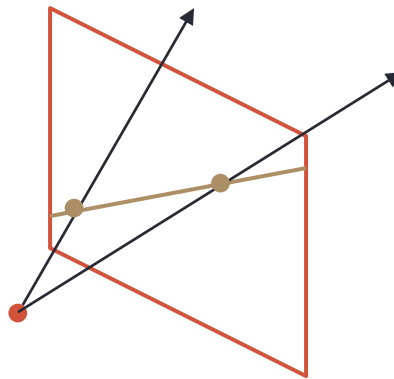


Last time: Homographies

- Provide mapping between images (image planes) taken from same center of projection; also mapping between any images of a planar surface.



Last time: Projective geometry



- A line is a *plane* of rays through origin
 - all rays (x,y,z) satisfying: $ax + by + cz = 0$

in vector notation :

$$0 = \begin{bmatrix} a & b & c \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

l p

- A line is also represented as a homogeneous 3-vector **l**

Projective Geometry: lines and points

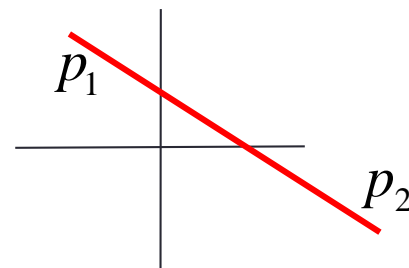
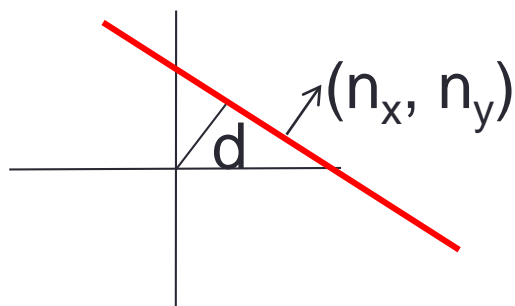
2D Lines: $ax + by + c = 0$

$$\begin{bmatrix} a & b & c \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = 0$$

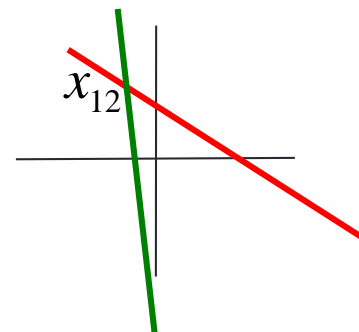
Eq of line

$$\mathbf{l}^T \mathbf{x} = 0$$

$$l = \begin{bmatrix} a & b & c \end{bmatrix} \Rightarrow \begin{bmatrix} n_x & n_y & d \end{bmatrix}$$



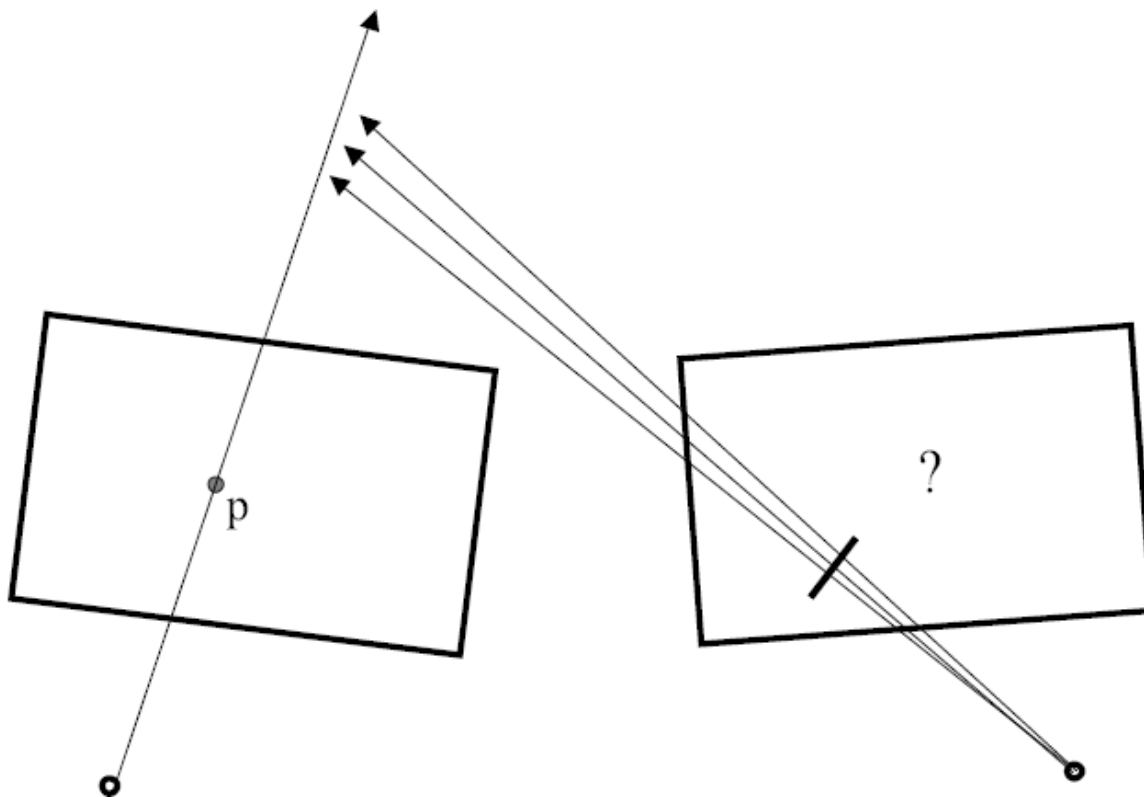
$$\left. \begin{array}{l} p_1 = \begin{bmatrix} x_1 & y_1 & 1 \end{bmatrix} \\ p_2 = \begin{bmatrix} x_2 & y_2 & 1 \end{bmatrix} \end{array} \right\} l = p_1 \times p_2$$



$$\left. \begin{array}{l} l_1 = \begin{bmatrix} a_1 & b_1 & c_1 \end{bmatrix} \\ l_2 = \begin{bmatrix} a_2 & b_2 & c_2 \end{bmatrix} \end{array} \right\} x_{12} = l_1 \times l_2$$

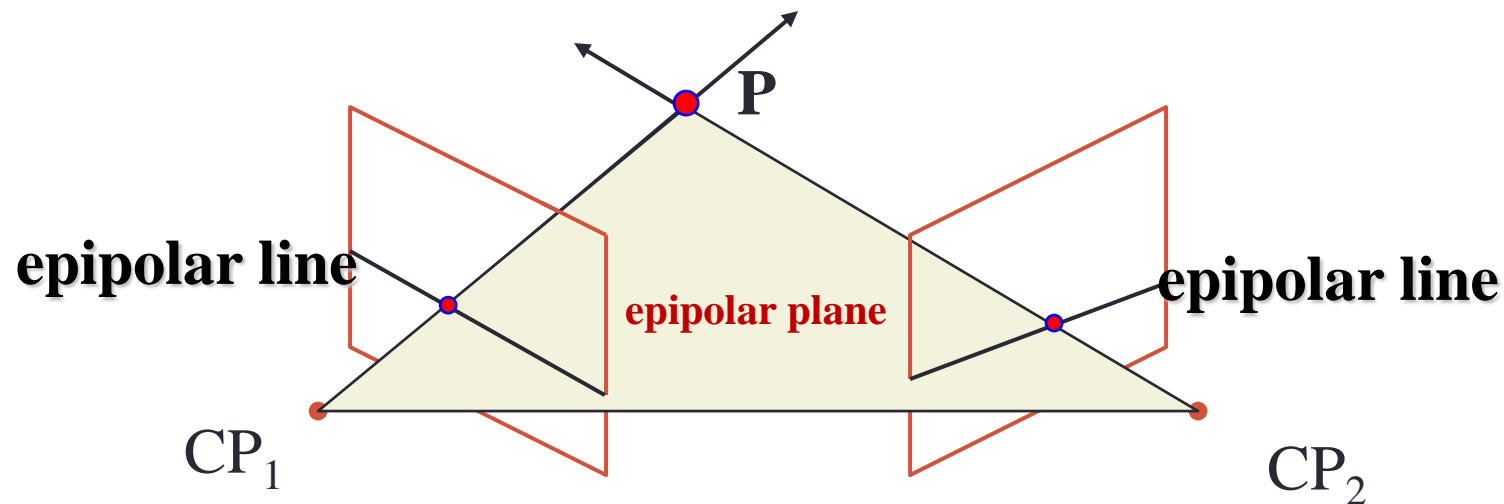
Motivating the problem: stereo

- Given two views of a scene (the two cameras not necessarily having optical axes) what is the relationship between the location of a scene point in one image and its location in the other?



Stereo correspondence

- Determine Pixel Correspondence
 - Pairs of points that correspond to same scene point

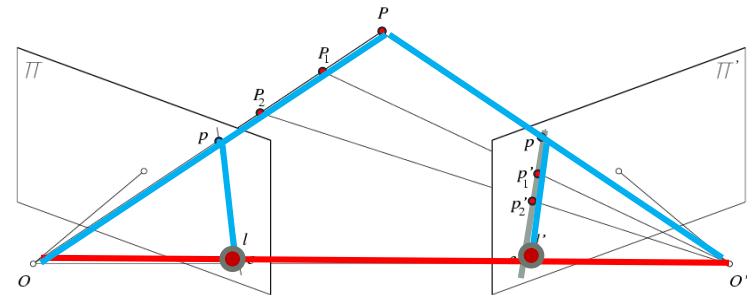


Epipolar Constraint

- Reduces correspondence problem to 1D search along *conjugate epipolar lines*

Epipolar geometry: terms

- **Baseline:** line joining the camera centers
 - **Epipole:** point of intersection of baseline with image plane
 - **Epipolar plane:** plane containing baseline and world point
 - **Epipolar line:** intersection of epipolar plane with the image plane
- All epipolar lines intersect at the epipole
 - An epipolar plane intersects the left and right image planes in epipolar lines



Example: converging cameras

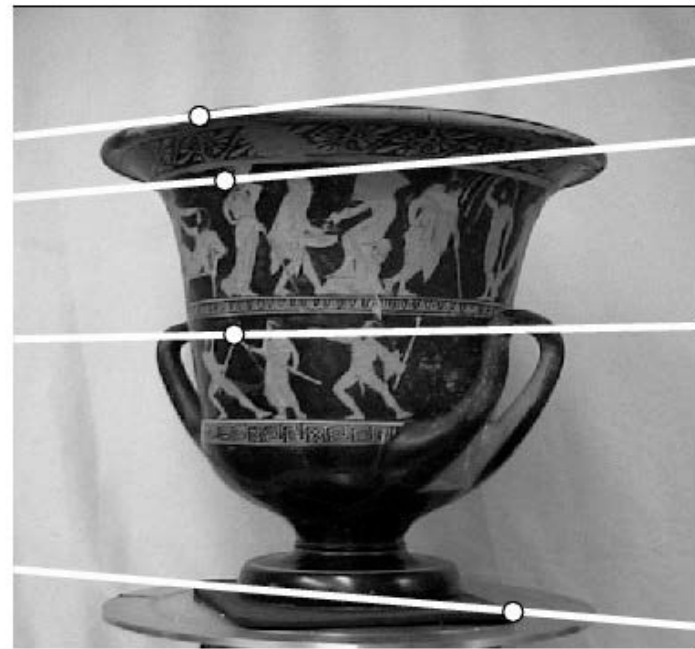
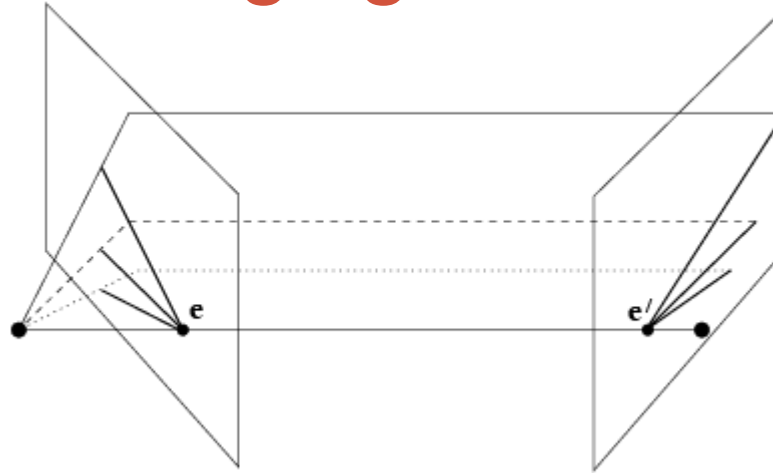
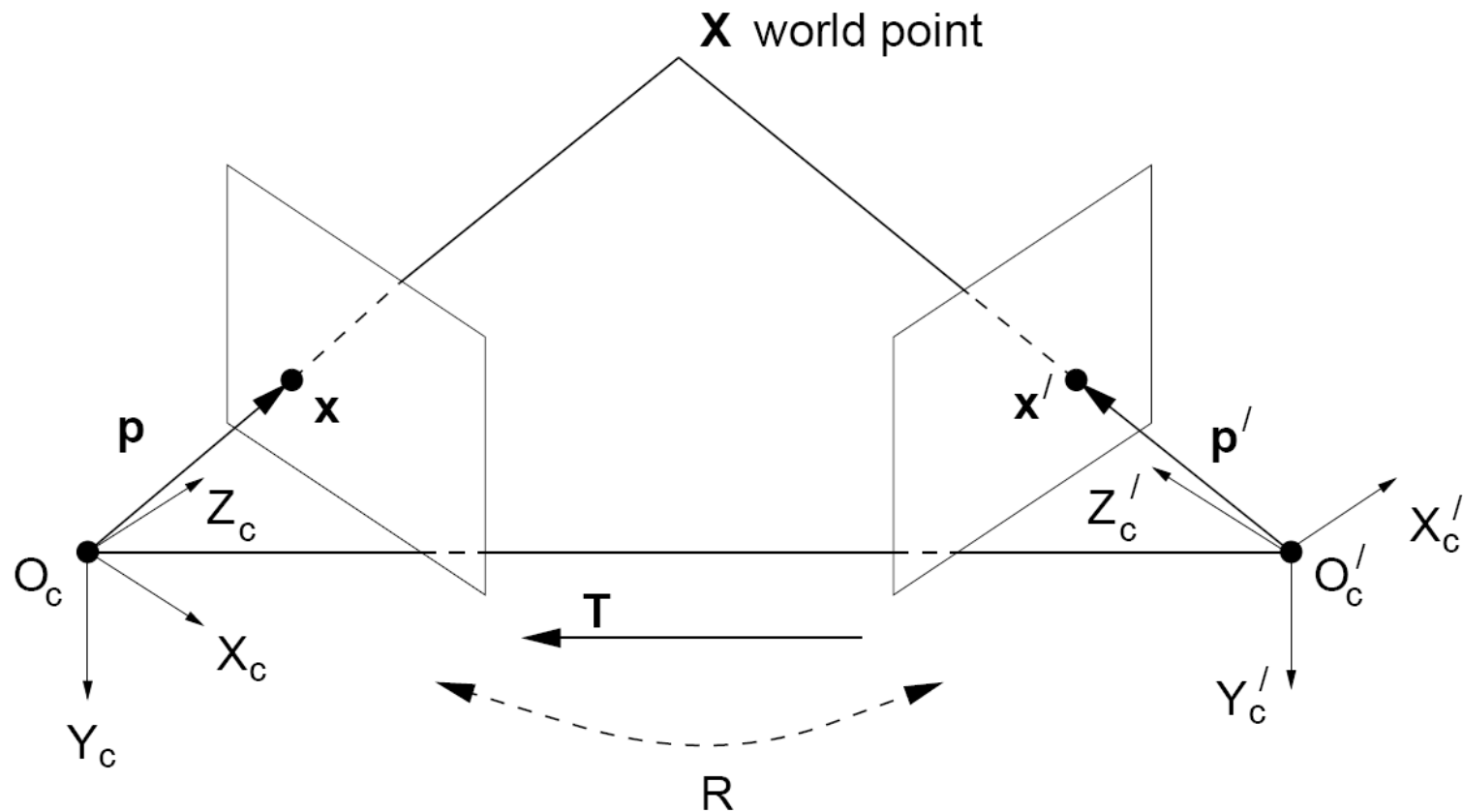


Figure from Hartley & Zisserman

From Geometry to Algebra

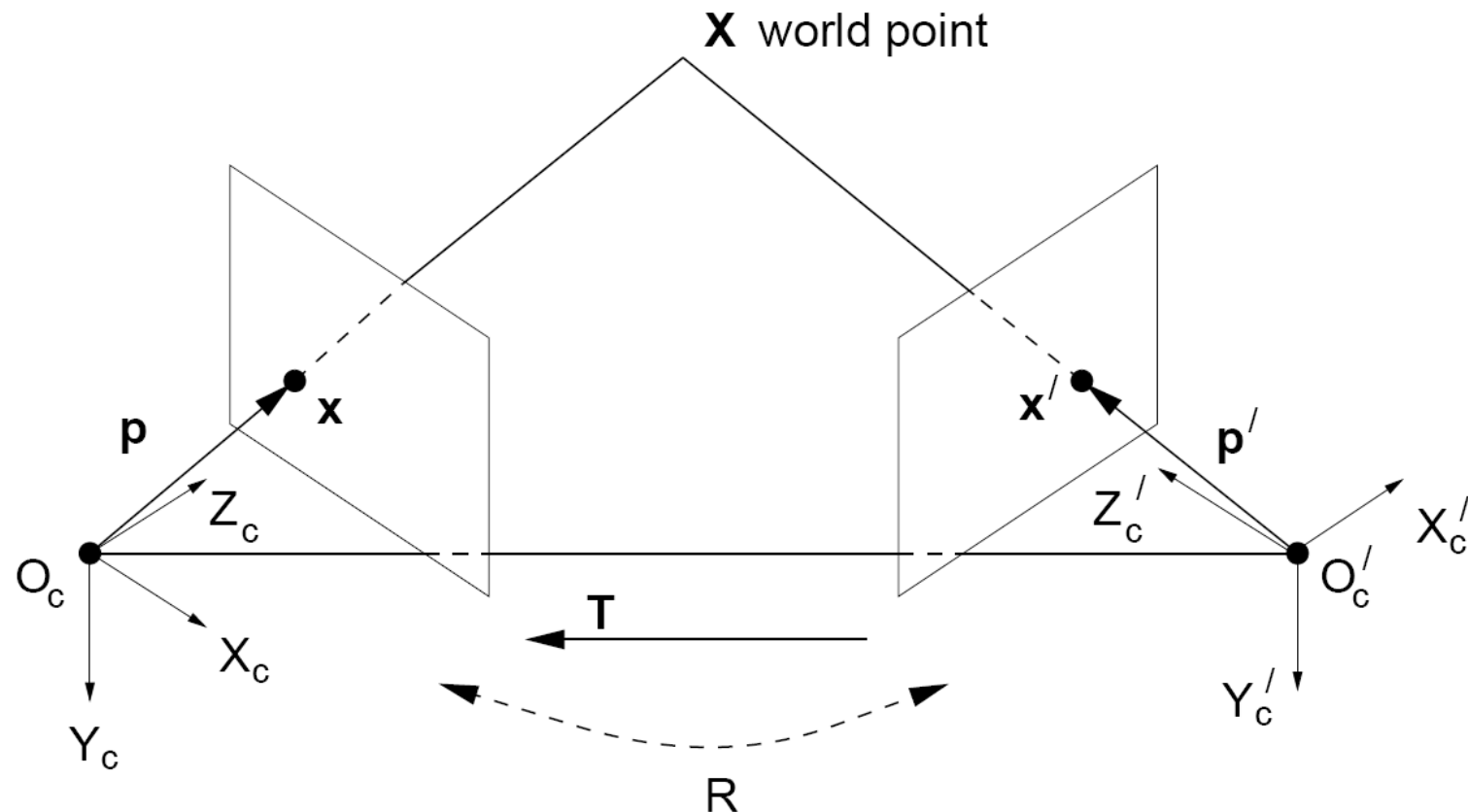
- So far, we have the explanation in terms of geometry.
- Now, how to express the epipolar constraints algebraically?

Stereo geometry, with calibrated cameras



Main idea

Stereo geometry, with calibrated cameras

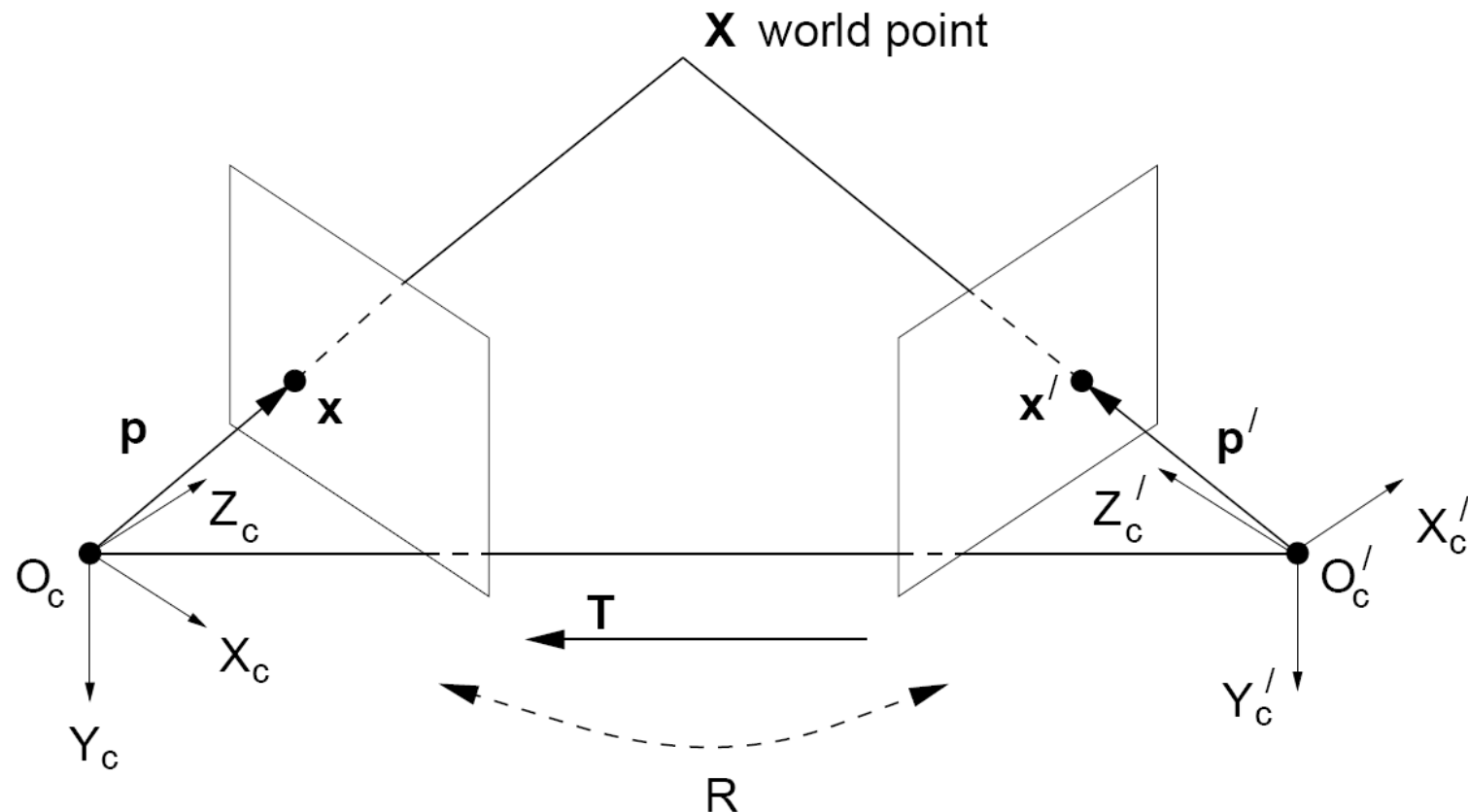


If the stereo rig is calibrated, we know :

how to **rotate** and **translate** camera reference frame 1 to get to camera reference frame 2.

Rotation: 3 x 3 matrix \mathbf{R} ; translation: 3 vector \mathbf{T} .

Stereo geometry, with calibrated cameras



If the stereo rig is calibrated, we know :

how to **rotate** and **translate** camera reference frame 1 to get to camera reference frame 2.

$$\mathbf{X}'_c = \mathbf{R}\mathbf{X}_c + \mathbf{T}$$

An aside: cross product

$$\vec{a} \times \vec{b} = \vec{c}$$

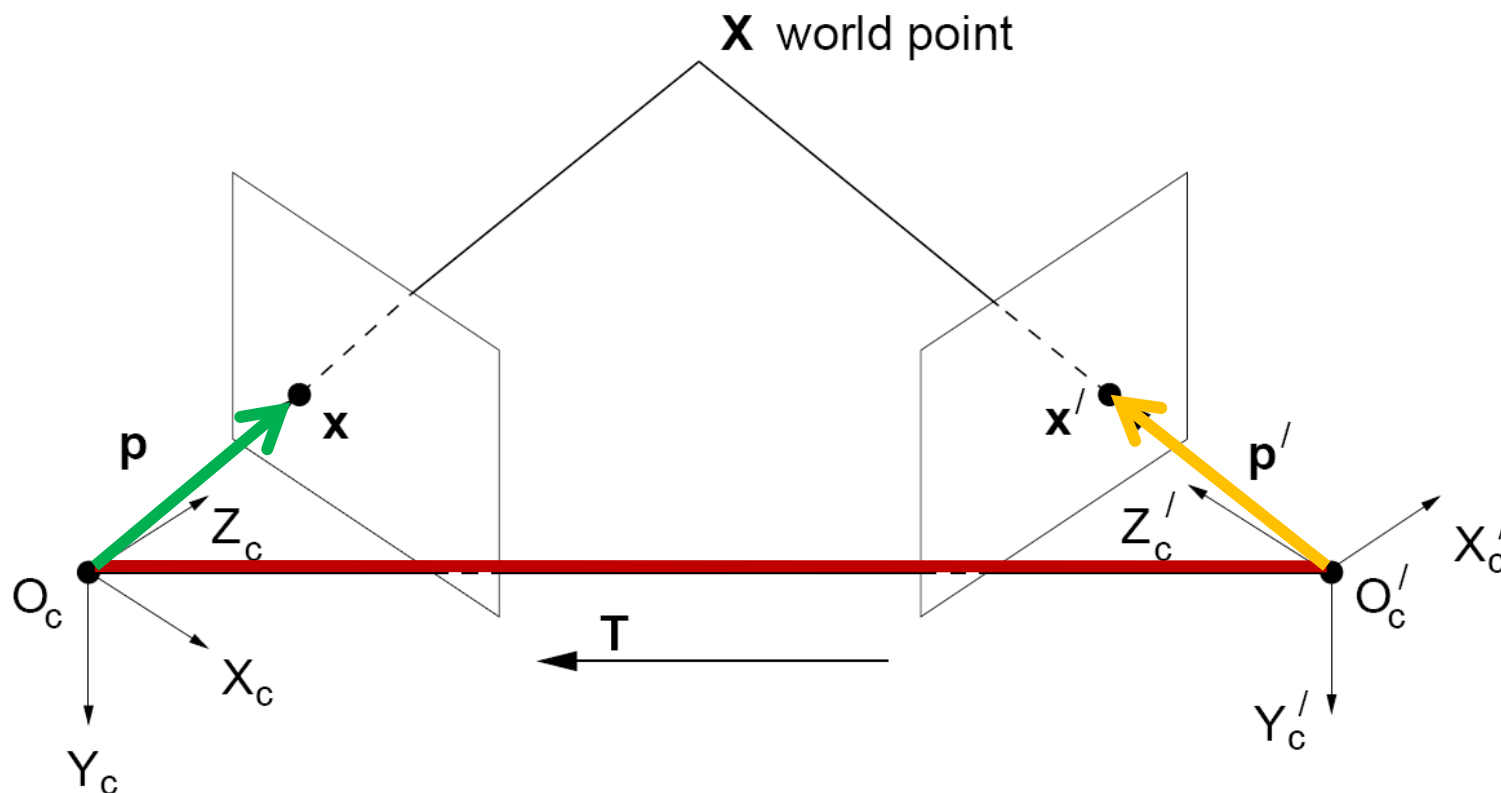
Vector cross product takes two vectors and returns a third vector that's perpendicular to both inputs.

So here, c is perpendicular to both a and b , which means the dot product $= 0$.

$$\vec{a} \cdot \vec{c} = 0$$

$$\vec{b} \cdot \vec{c} = 0$$

From geometry to algebra



$$\mathbf{X}' = \mathbf{R}\mathbf{X} + \mathbf{T}$$

$$\underbrace{\mathbf{T} \times \mathbf{X}'}_{\text{Normal to the plane}} = \mathbf{T} \times \mathbf{R}\mathbf{X}$$

$$\mathbf{X}' \cdot (\mathbf{T} \times \mathbf{X}') = \mathbf{X}' \cdot (\mathbf{T} \times \mathbf{R}\mathbf{X})$$

$$0 = \mathbf{X}' \cdot (\mathbf{T} \times \mathbf{R}\mathbf{X})$$

Another aside: Matrix form of cross product

$$\vec{a} \times \vec{b} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \vec{c}$$

Can be expressed as a matrix multiplication!!!

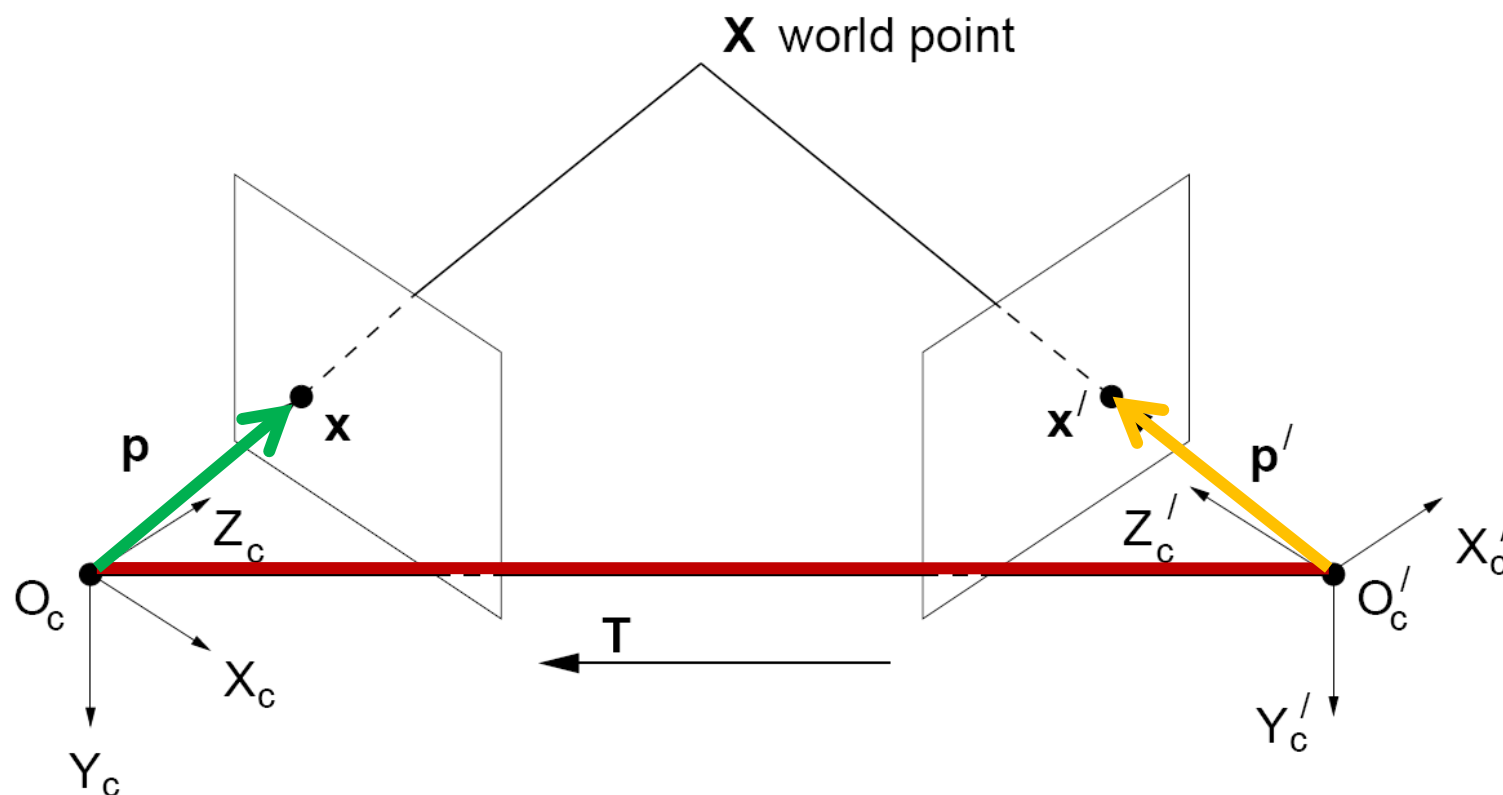
$$[a_x] = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}$$

Notation:

$$\vec{a} \times \vec{b} = [\vec{a}_\times] \vec{b}$$

Has rank 2!

From geometry to algebra



$$\mathbf{X}' = \mathbf{R}\mathbf{X} + \mathbf{T}$$

$$\underbrace{\mathbf{T} \times \mathbf{X}'}_{\text{Normal to the plane}} = \mathbf{T} \times \mathbf{R}\mathbf{X}$$

$$\mathbf{X}' \cdot (\mathbf{T} \times \mathbf{X}') = \mathbf{X}' \cdot (\mathbf{T} \times \mathbf{R}\mathbf{X})$$

$$0 = \mathbf{X}' \cdot (\mathbf{T} \times \mathbf{R}\mathbf{X})$$

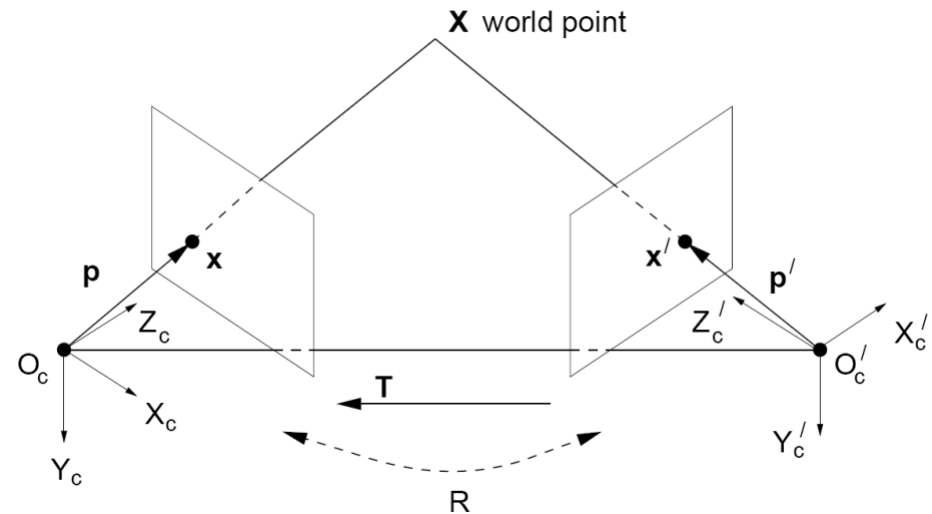
Essential matrix

$$\mathbf{X}' \cdot (\mathbf{T} \times \mathbf{R}\mathbf{X}) = 0$$

$$\mathbf{X}' \cdot ([\mathbf{T}_x] \mathbf{R}\mathbf{X}) = 0$$

Let $\mathbf{E} = [\mathbf{T}_x] \mathbf{R}$

$$\mathbf{X}'^T \mathbf{E} \mathbf{X} = 0$$

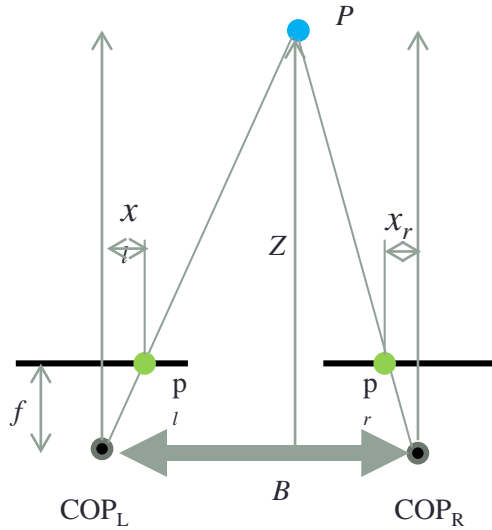


\mathbf{E} is called the **essential matrix**, and it relates corresponding image points between both cameras, given the rotation and translation.

Note: these points are in **each camera coordinate systems**.

We know if we observe a point in one image, its position in other image is constrained to lie on line defined by above.

Essential matrix example: parallel cameras

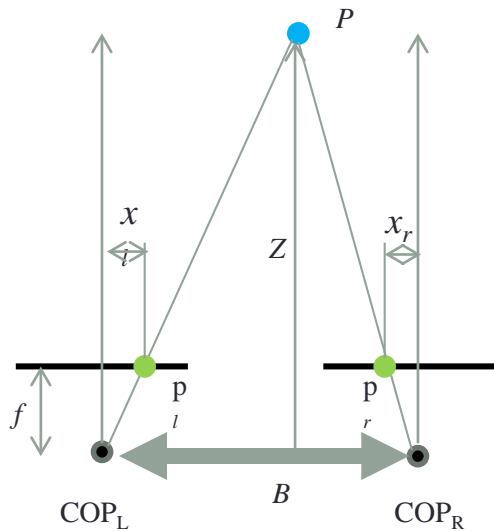


$$\mathbf{R} =$$

$$\mathbf{T} =$$

$$\mathbf{E} = [\mathbf{T}_x] \mathbf{R} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & B \\ 0 & -B & 0 \end{pmatrix}$$

Essential matrix example: parallel cameras



$$\mathbf{E} = [\mathbf{T}_x] \mathbf{R} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & B \\ 0 & -B & 0 \end{pmatrix}$$

$$\mathbf{p}'^T \mathbf{E} \mathbf{p} = 0 : \begin{bmatrix} x' & y' & f \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & B \\ 0 & -B & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ f \end{bmatrix} = 0$$

$$\mathbf{p} = [X, Y, Z] = \left[\frac{Zx}{f}, \frac{Zy}{f}, Z \right] \cong [x, y, f]$$

$$\mathbf{p}' = [X', Y', Z] = \left[\frac{Zx'}{f}, \frac{Zy'}{f}, Z \right] \cong [x', y', f]$$

$$\begin{bmatrix} x' & y' & f \end{bmatrix} \begin{bmatrix} 0 \\ Bf \\ -By \end{bmatrix} = 0$$

$$Bfy' = Bfy \Rightarrow \mathbf{y}' = \mathbf{y}$$

**For the parallel cameras,
image of any point must lie
on same horizontal line in
each image plane.**

Given a known point (x,y) in the original image, this is a *line* in the (x',y') image.

Weak calibration

- Want to estimate world geometry without requiring calibrated cameras
 - Archival videos (already have the pictures)
 - Photos from multiple unrelated users
 - Dynamic camera system
- **Main idea:**
 - Estimate epipolar geometry from a (redundant) set of point correspondences between two uncalibrated cameras

From before: Projection matrix

- This can be rewritten as a matrix product using homogeneous coordinates:

where:

$$\begin{bmatrix} s x_{im} \\ s y_{im} \\ s \end{bmatrix} = \mathbf{K}_{int} \mathbf{\Phi}_{ext} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

$$\mathbf{\Phi}_{ext} = \begin{bmatrix} \mathbf{r}_{11} & \mathbf{r}_{12} & \mathbf{r}_{13} & -\mathbf{R}_1^T \mathbf{T} \\ \mathbf{r}_{21} & \mathbf{r}_{22} & \mathbf{r}_{23} & -\mathbf{R}_2^T \mathbf{T} \\ \mathbf{r}_{31} & \mathbf{r}_{32} & \mathbf{r}_{33} & -\mathbf{R}_3^T \mathbf{T} \end{bmatrix}$$

$$\mathbf{K}_{int} = \begin{bmatrix} -f / s_x & 0 & o_x \\ 0 & -f / s_y & o_y \\ 0 & 0 & 1 \end{bmatrix}$$

Note: Invertible, scale x and y , assumes no skew

From before: Projection matrix

- This can be rewritten as a matrix product using homogeneous coordinates:

$$\begin{bmatrix} s x_{im} \\ s y_{im} \\ s \end{bmatrix} = \mathbf{K}_{int} \mathbf{\Phi}_{ext} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

$$\mathbf{p}_{im} = \mathbf{K}_{int} \underbrace{\mathbf{\Phi}_{ext} \mathbf{P}_w}_{\mathbf{p}_c}$$

$$\mathbf{p}_{im} = \mathbf{K}_{int} \mathbf{p}_c$$

Uncalibrated case

For a given
camera:

$$\mathbf{p}_{im} = \mathbf{K}_{int} \mathbf{p}_c$$

So, for **two** cameras (left and right):

$$\mathbf{p}_{c,left} = \mathbf{K}_{int,left}^{-1} \mathbf{p}_{im,left}$$

$$\mathbf{p}_{c,right} = \underbrace{\mathbf{K}_{int,right}^{-1}}_{\text{Internal calibration matrices, one per camera}} \mathbf{p}_{im,right}$$

Internal calibration
matrices, one per
camera

Uncalibrated case

$$\mathbf{p}_{c,right} = \mathbf{K}_{int,right}^{-1} \mathbf{p}_{im,right}$$

$$\mathbf{p}_{c,left} = \mathbf{K}_{int,left}^{-1} \mathbf{p}_{im,left}$$

From before, the **essential**
matrix \mathbf{E} .

$$\mathbf{p}_{c,right}^T \mathbf{E} \mathbf{p}_{c,left} = 0$$

$$\left(\mathbf{K}_{int,right}^{-1} \mathbf{p}_{im,right} \right)^T \mathbf{E} \left(\mathbf{K}_{int,left}^{-1} \mathbf{p}_{im,left} \right) = 0$$

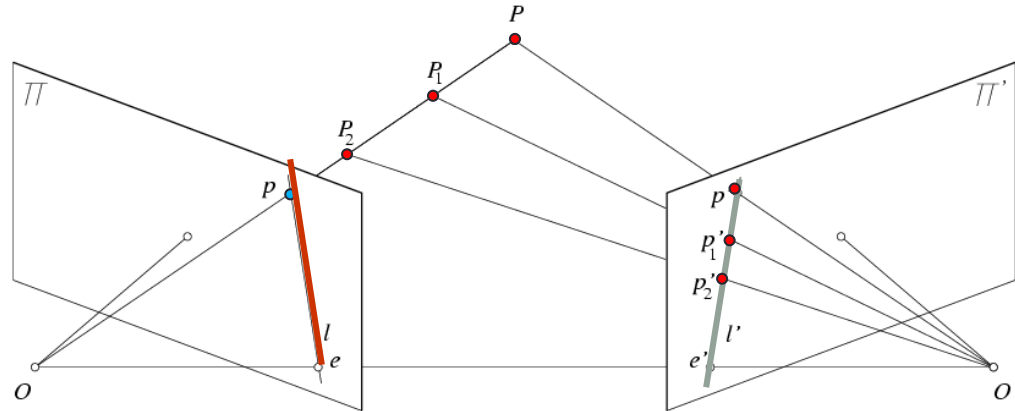
$$\mathbf{p}_{im,right}^T \underbrace{\left(\left(\mathbf{K}_{int,right}^{-1} \right)^T \mathbf{E} \mathbf{K}_{int,left}^{-1} \right)}_{\mathbf{F}} \mathbf{p}_{im,left} = 0$$

“Fundamental matrix” \mathbf{F}

$$\mathbf{p}_{im,right}^T \mathbf{F} \mathbf{p}_{im,left} = 0 \quad \text{or} \quad \mathbf{p}^T \mathbf{F} \mathbf{p}' = 0$$

Properties of the Fundamental Matrix

$$\mathbf{p}^T \mathbf{F} \mathbf{p}' = 0$$



- $\mathbf{l} = \mathbf{F} \mathbf{p}'$ is the epipolar line associated with \mathbf{p}'
- $\mathbf{l}' = \mathbf{F}^T \mathbf{p}$ is the epipolar line associated with \mathbf{p}
- Epipoles found by $\mathbf{F} \mathbf{p}' = \mathbf{0}$ and $\mathbf{F}^T \mathbf{p} = \mathbf{0}$ (Why?)
- \mathbf{F} is singular (mapping from 2-D point to 1-D family so rank 2 – more later)

Fundamental matrix

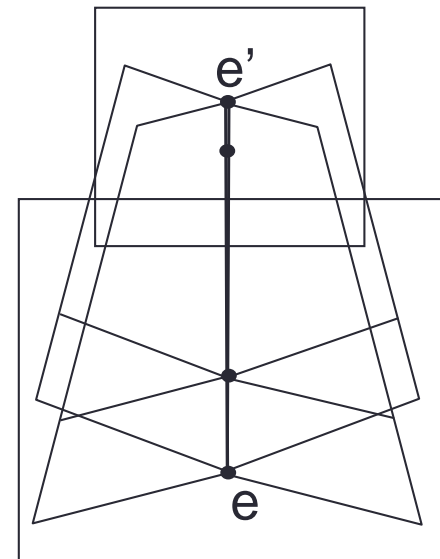
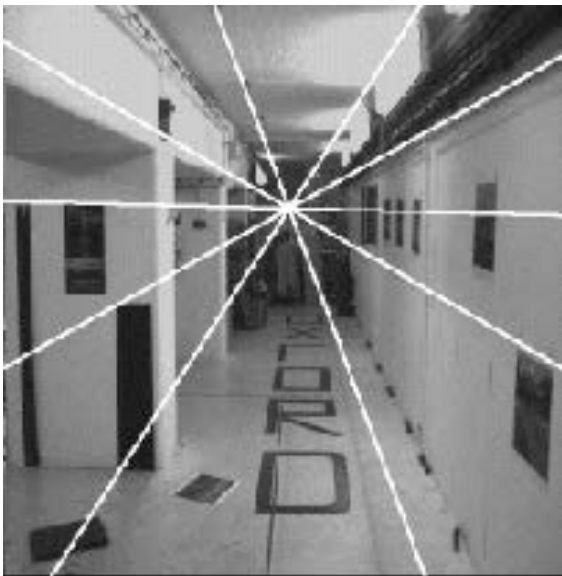
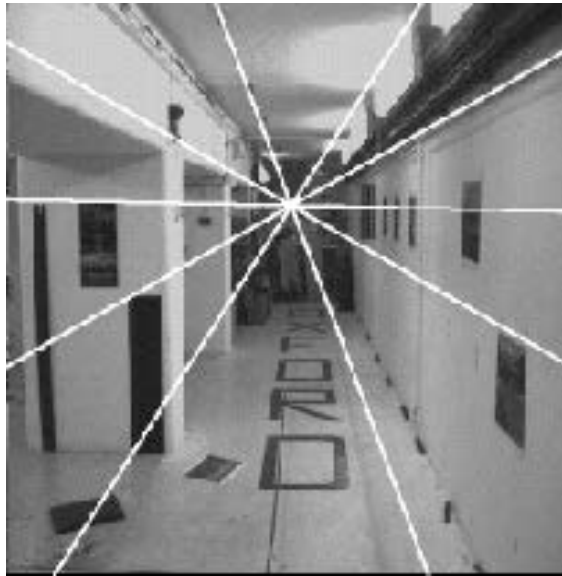
- Relates pixel coordinates in the two views
- More general form than essential matrix: we remove need to know intrinsic parameters
- If we estimate fundamental matrix from correspondences in pixel coordinates, can reconstruct epipolar geometry without intrinsic or extrinsic parameters.



Any two views?

- Can we do this for any two views?
- Yes, as long as can see some similar points between images?
- So how about these...

Different Example: forward motion



courtesy of Andrew Zisserman

Computing F from correspondences

Each point
correspondence
generates one
constraint on F

$$\mathbf{p}_{im,right}^T \mathbf{F} \mathbf{p}_{im,left} = 0$$

$$\begin{bmatrix} u' & v' & 1 \end{bmatrix} \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = 0$$

Collect n of these
constraints

$$\begin{bmatrix} u'_1 u_1 & u'_1 v_1 & u'_1 & v'_1 u_1 & v'_1 v_1 & v'_1 & u_1 & v_1 & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = \mathbf{0}$$

Solve for f , vector of parameters.

The (in)famous “eight-point algorithm”

250906.36	183269.57	921.81	200931.10	146766.13	738.21	272.19	198.81	1.00
2692.28	131633.03	176.27	6196.73	302975.59	405.71	15.27	746.79	1.00
416374.23	871684.30	935.47	408110.89	854384.92	916.90	445.10	931.81	1.00
191183.60	171759.40	410.27	416435.62	374125.90	893.65	465.99	418.65	1.00
48988.86	30401.76	57.89	298604.57	185309.58	352.87	846.22	525.15	1.00
164786.04	546559.67	813.17	1998.37	6628.15	9.86	202.65	672.14	1.00
116407.01	2727.75	138.89	169941.27	3982.21	202.77	838.12	19.64	1.00
135384.58	75411.13	198.72	411350.03	229127.78	603.79	681.28	379.48	1.00

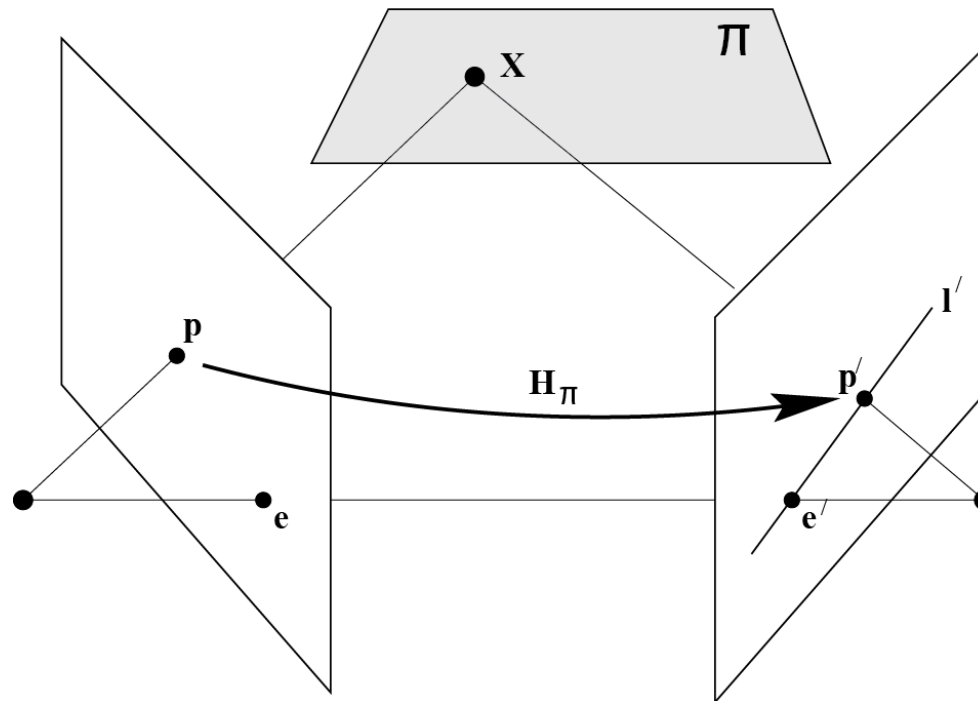
$$\begin{pmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{pmatrix} = 0$$

- In principal can solve with 8 points.
- Better with more – yields homogeneous linear least-squares:
 - Find unit norm vector F yielding smallest residual
 - Remember SVD or substitute a 1?
- What happens when there is noise?

Doing the obvious thing



Rank of F



- Assume we know the homography H_π that maps from Left to Right (Full 3x3)

$$\mathbf{p}' = \mathbf{H}_\pi \mathbf{p}$$

- Let line l' be the epipolar line corresponding to \mathbf{p} – goes through epipole \mathbf{e}'

$$\begin{aligned} \text{So: } \mathbf{l}' &= \mathbf{e}' \times \mathbf{p}' \\ &= \mathbf{e}' \times \mathbf{H}_\pi \mathbf{p} \\ &= [\mathbf{e}']_\times \mathbf{H}_\pi \mathbf{p} \\ &= \mathbf{F} \mathbf{p} \end{aligned}$$

- Rank of \mathbf{F} is rank of $[\mathbf{e}']_\times = 2$

Fix the linear solution

- Use SVD or other method to do linear computation for F
- Decompose F using SVD (not the same SVD):

$$\mathbf{F} = \mathbf{U}\mathbf{D}\mathbf{V}^T$$

- Set the last singular value to zero:

$$\mathbf{D} = \begin{bmatrix} r & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & t \end{bmatrix} \Rightarrow \hat{\mathbf{D}} = \begin{bmatrix} r & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

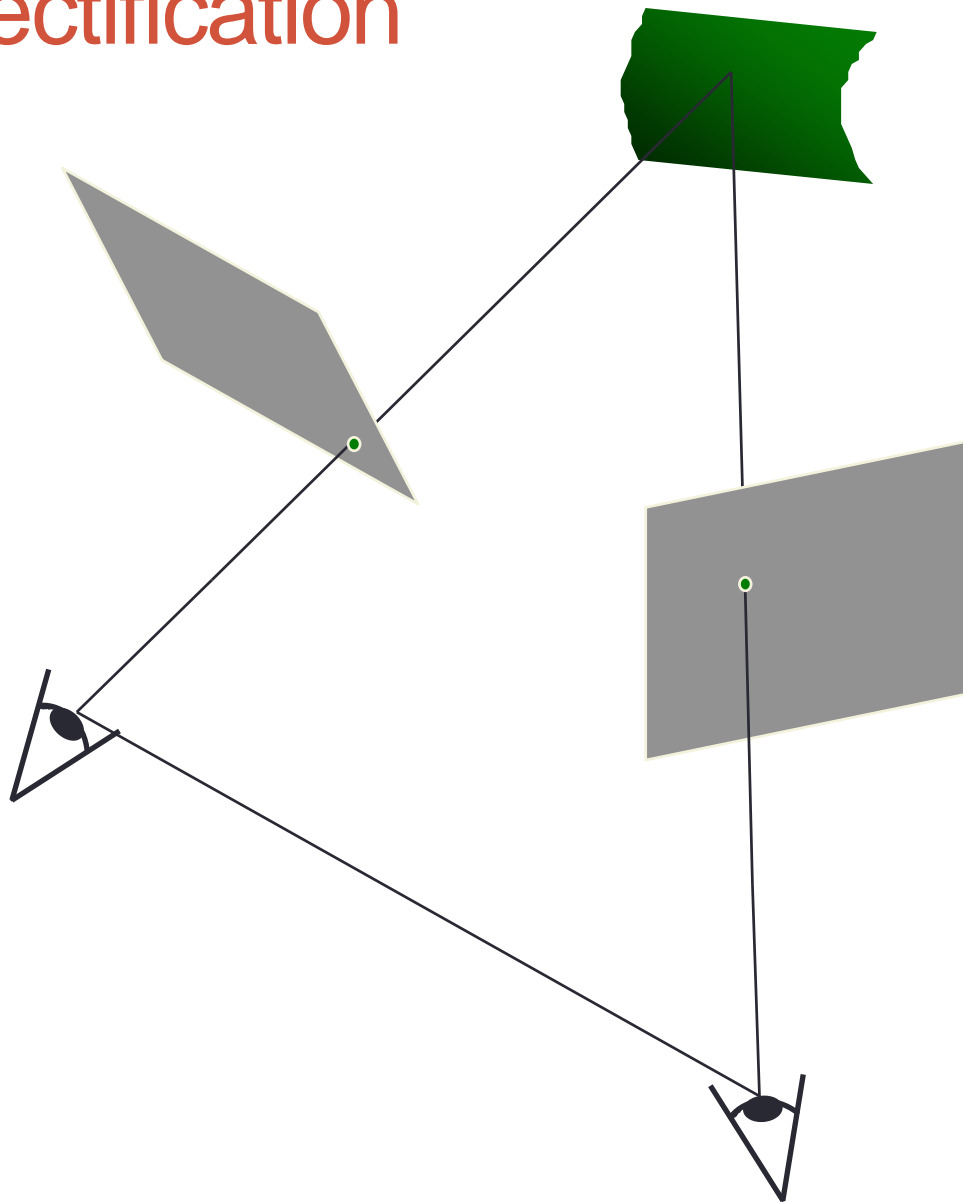
- Estimate new F from the new $\hat{\mathbf{D}}$

$$\hat{\mathbf{F}} = \mathbf{U}\hat{\mathbf{D}}\mathbf{V}^T$$

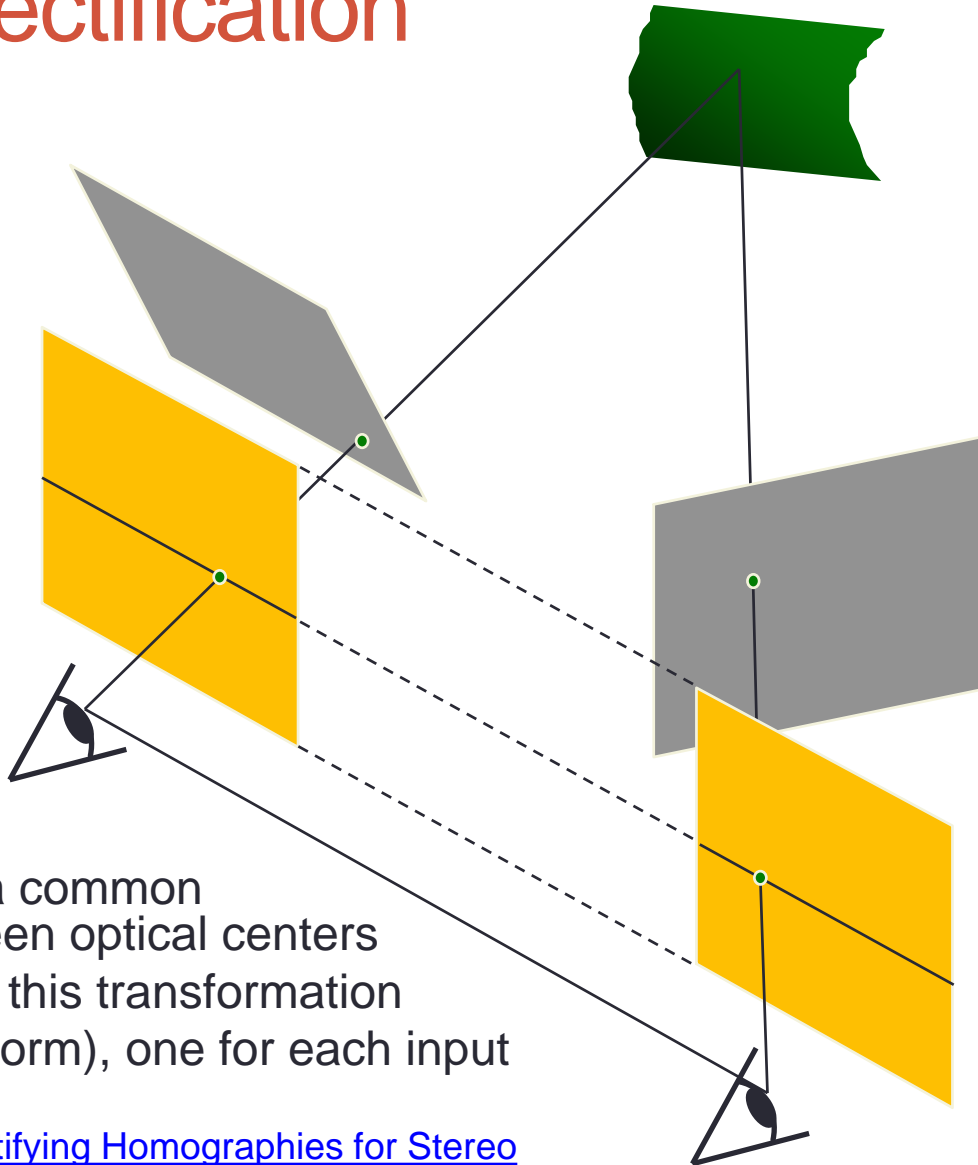
That's better...



Stereo image rectification



Stereo image rectification



- Reproject image planes onto a common plane parallel to the line between optical centers
 - Pixel motion is horizontal after this transformation
 - Two homographies (3x3 transform), one for each input image reprojection
- C. Loop and Z. Zhang. [Computing Rectifying Homographies for Stereo Vision](#). IEEE Conf. Computer Vision and Pattern Recognition, 1999.

Rectification Example

C. Loop and Z. Zhang,
Computing Rectifying
Homographies for
Stereo Vision,
IEEE Conf. Computer
Vision and Pattern
Recognition, 1999.



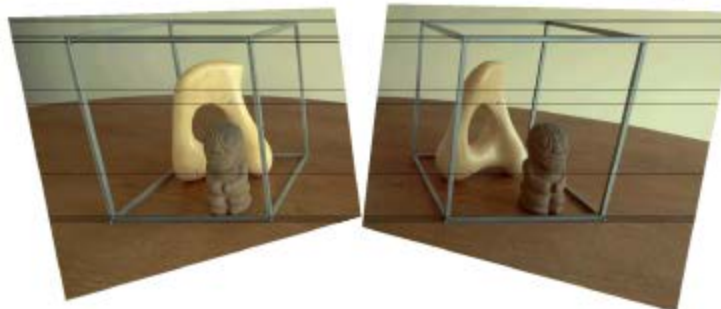
(a) Original image pair overlaid with several epipolar lines.



(b) Image pair transformed by the specialized projective mapping H_l and H'_l . Note that the epipolar lines are now parallel to each other in each image.



(c) Image pair transformed by the similarity H_l and H'_l . Note that the image pair is now rectified (the epipolar lines are horizontally aligned).

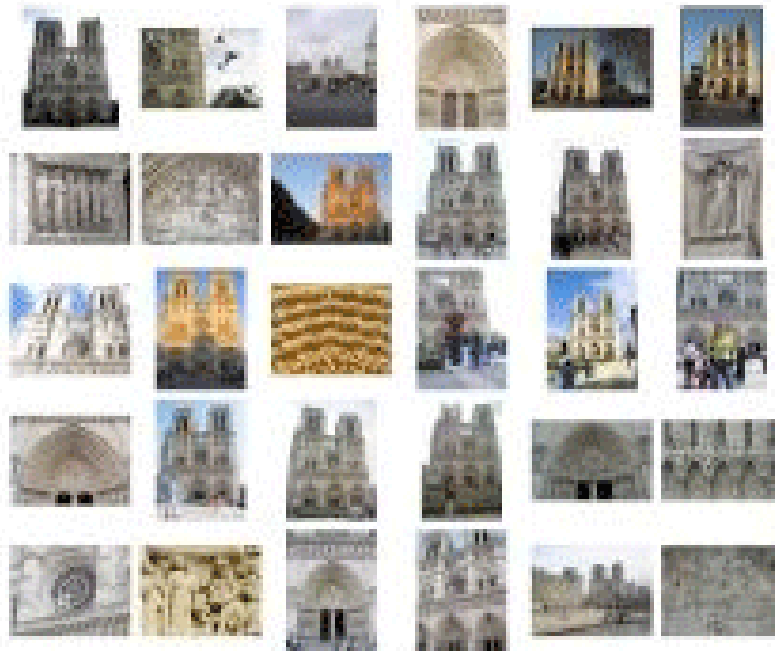


(d) Final image rectification after shearing transform H_l and H'_l . Note that the image pair remains rectified, but the horizontal distortion is reduced.

Some example cool applications...

Photo synth

Noah Snavely, Steven M. Seitz, Richard Szeliski, "[Photo tourism: Exploring photo collections in 3D](#)," SIGGRAPH 2006



<http://photosynth.net/>

Photosynth.net

The screenshot shows the Photosynth.net website. At the top left is the Microsoft Live Labs logo and the Photosynth logo. At the top right are links for 'Sign In' and a search bar labeled 'Search Synths'. The main content area features a large central image titled 'Inauguration of the 44th President' with the text '613 photos, 58% synthy' below it. Below this is a horizontal carousel of image thumbnails. The fifth thumbnail, showing the Great Pyramid of Giza, is highlighted with a green border and a 'Go' button. Below the carousel, the text reads 'Great Pyramid of Giza HDR by mokojo100' and '310 Photos - 95% Synthy'. At the bottom, there are five icons with corresponding text: a camera for 'Create your Synth', a leaf for 'About Photosynth', a collage of buildings for 'Explore Synths', a document for 'Latest Synth News', and speech bubbles for 'Discussion Forum'. A green button at the bottom center says 'Read the latest Photosynth news and updates on the blog'. The footer contains copyright information and links for 'Live Search', 'MSN', 'Windows Live', 'Privacy', 'Legal', 'Microsoft Live Labs', 'Help', 'Blog', and 'Contact Us'.

Microsoft® Live Labs™

Photosynth™

Sign In | Search Synths

Inauguration of the 44th President

613 photos, 58% synthy

Go

Great Pyramid of Giza HDR by mokojo100
310 Photos - 95% Synthy

Create your Synth

About Photosynth

Explore Synths

Latest Synth News

Discussion Forum

Read the latest Photosynth news and updates on the blog

© Microsoft Corporation | Live Search | MSN | Windows Live | Privacy | Legal

Microsoft Live Labs | Help | Blog | Contact Us

Based on [Photo Tourism](#)
by Noah Snavely, Steve Seitz, and Rick Szeliski

3D from multiple images



Building Rome in a Day: Agarwal et al. 2009

Summary

- For 2-views, there is a geometric relationship that define the relations between rays in one view to rays in the other
 - Calibrated – Essential matrix
 - Uncalibrated – Fundamental matrix.
- This relation can be estimated from point correspondences – both in calibrated cases and uncalibrated.
- Extensions allow combining multiple views to get more geometric information about scenes
 - SLAM (simultaneous localization and mapping) – you'll hear about this if we fit it in...